In the Claims:

Claims 41-67 are pending as listed below:

1-40. (Cancelled)

41. (Previously presented) A method for accessing a memory, comprising:

comparing a memory address of a memory access request to decompressed defective memory addresses that are otherwise stored in a compressed format, the defective memory addresses having substitute addresses associated and stored therewith;

where the memory address matches one of the decompressed defective memory addresses, extracting the substitute address associated therewith; and

accessing a memory location corresponding to the extracted substitute address rather than a memory location corresponding to the memory address.

42. (Previously presented) The method of claim 41 wherein comparing the memory address of the memory access request comprises:

decompressing a portion of at least one of the stored defective memory address;

calculating a value from the memory address; and

comparing the calculated value to the decompressed portion.

43. (Previously presented) The method of claim 42, further comprising:

where the calculated value fails to match the decompressed portion of the defective memory address, decompressing a portion of another one of the stored defective memory addresses; and

comparing the calculated value to the most recently decompressed portion.

44. (Previously presented) The method of claim 42 wherein calculating a value from the memory address comprises dividing the value represented by the memory address by a prime number.

45. (Previously presented) A method for accessing a memory device receiving memory addresses, the method comprising:

comparing the received memory addresses to decompressed addresses of defective memory locations in the memory device, the decompressed addresses of the defective memory locations otherwise stored in a compressed format having associated therewith substitute addresses corresponding to substitute memory locations in another memory; and

substituting for the memory addresses matching the decompressed addresses of defective memory locations the associated substitute memory addresses to access the substitute memory locations in the other memory.

46. (Previously presented) The method of claim 45 wherein the decompressed addresses of defective memory locations in the memory device are stored in a compressed format and wherein comparing the received memory addresses comprises decompressing at least one of the stored addresses of defective memory locations, and comparing a received memory address to the decompressed address.

47. (Previously presented) The method of claim 45 wherein the decompressed addresses of defective memory locations in the memory device are stored in a compressed format and wherein comparing the received memory addresses comprises:

decompressing a portion of at least one of the stored addresses of defective memory locations;

calculating a value from a received memory address; and

comparing the calculated value to the decompressed portion.

48. (Previously presented) The method of claim 47, further comprising:

where the calculated value fails to match the decompressed portion of the stored address of a defective memory location, decompressing a portion of another one of the stored addresses of defective memory locations; and

comparing the calculated value to the most recently decompressed portion.

49. (Previously presented) A method for accessing a requested memory location of a memory array, the requested memory location having a requested address, the method comprising:

generating a first hash code from the requested address;

comparing the first hash code to hash codes for decompressed addresses stored in a temporary memory array;

when a match is found between a hash code for a decompressed address and the first hash code, determining if an address stored in the temporary array corresponds to the requested address; and

accessing a spare memory array when an address stored in the temporary array corresponds to the requested address.

50. (Previously presented) The method of claim 49 wherein generating the first hash code comprises dividing the value represented by the requested address by a prime number.

51. (Previously presented) The method of claim 49 wherein the memory array and the spare memory array are separate memory devices.

52. (Previously presented) The method of claim 49 wherein determining if an address stored in the temporary array corresponds to the requested address comprises comparing the requested address with decompressed addresses having the same hash code until a match is made.

53. (Previously presented) The method of claim 49 wherein the memory array and spare memory array comprise two separate memory devices.

54. (Previously presented) A method for storing memory addresses of defective cells in a memory array, the method comprising:

receiving a memory test command;

initiating a memory test in response to the memory test command, the memory test for determining memory addresses of defective memory cells of the memory array;

mapping memory addresses of defective memory cells to substitute addresses of substitute memory cells; and

compressing the memory addresses of defective memory cells and the substitute addresses associated therewith.

55. (Previously presented) The method of claim 54 wherein initiating the memory test comprises storing addresses of defective memory cells in a temporary memory array.

56. (Previously presented) The method of claim 54, further comprising storing the compressed address data in a map memory array.

57. (Previously presented) The method of claim 54, further comprising generating a unique code word for each of the addresses of the defective memory cells.

58. (Previously presented) The method of claim 54, further comprising issuing the memory test command upon initially powering-up the memory array.

59. (Previously presented) A method of remapping defective memory locations of a primary memory, the method comprising:

identifying memory addresses of the defective memory locations in the primary memory;

mapping the identified memory addresses of the primary memory to substitute memory addresses that correspond to substitute memory locations in a spare memory;

storing the identified memory addresses of the primary memory and the substitute memory addresses of the spare memory; and

in response to a request to access a defective memory location in the primary memory, substituting the associated substitute memory address in the spare memory for the memory address corresponding to the requested defective memory location in the primary memory.

60. (Previously presented) The method of claim 59 wherein the primary memory having the defective memory locations comprises a first memory device and the spare memory comprises a second memory device.

61. (Previously presented) The method of claim 59 wherein identifying the defective memory locations comprises testing all of the memory locations of the primary memory prior to accepting a first memory access request.

62. (Previously presented) The method of claim 59, further comprising compressing the defective memory addresses and the substitute addresses prior to storing.

63. (Previously presented) The method of claim 62 wherein compressing the defective memory addresses comprises generating a unique code word for each defective memory address.

64. (Previously presented) A method for accessing memory locations in a memory array, the method comprising steps of:

determining whether a memory address matches an address of a defective memory cell, the addresses stored in a temporary memory array;

where a match is determined, accessing a substitute memory location corresponding to a substitute memory address associated and stored with the matching address, the substitute memory location located in a separate memory array; and

otherwise, access the memory location in the memory array corresponding to the memory address.

65. (Previously presented) The method of claim 64 wherein determining whether the memory address matches comprises:

analyzing the memory address to determine which portion of compressed data stored in a map memory array containing compressed addresses of defective cells in a first memory array to decompress;

decompressing the portion of compressed data to provide expanded data;

writing the expanded data to the temporary memory array; and

comparing the expanded data to the memory address to determine whether the address corresponds to an expanded datum of the expanded data, when the address and the expanded datum match.

66. (Previously presented) The method of claim 65 wherein analyzing the memory address comprises:

generating a hash code from the memory address; and

comparing the hash code to hash codes identifying each portion of compressed data stored in the map memory array;

67. (Previously presented) The method of claim 66 wherein generating a hash code from the memory address comprises dividing the value represented by the memory address by a prime number.